LCA Methodology: Uncertainties in LCA

Uncertainty Calculation in Life Cycle Assessments

A Combined Model of Simulation and Approximation

Andreas Ciroth*[1], Günter Fleischer[2] and Jörg Steinbach[3]

[1] GreenDeltaTC GmbH, Raumerstr. 7, D-10437 Berlin, Germany
[2] TU Berlin, Institute of Environmental Technology, CR2, Jebensstr. 1, D-10623 Berlin, Germany
[3] TU Berlin, Institute of Process and Plant Technology (IPAT), TK031, Str. des 17. Juni 135, D-10623 Berlin, Germany

* Corresponding author (ciroth@greendeltatc.com)

DOI: http://dx.doi.org/10.1065/lca2004.05.158

Abstract

Goal and Background. Uncertainty is commonly not taken into account in LCA studies, which downgrades their usability for decision support. One often stated reason is a lack of method. The aim of this paper is to develop a method for calculating the uncertainty propagation in LCAs in a fast and reliable manner.

Approach. The method is developed in a model that reflects the calculation of an LCA. For calculating the uncertainty, the model combines approximation formulas and Monte Carlo Simulation. It is based on virtual data that distinguishes true values and random errors or uncertainty, and that hence allows one to compare the performance of error propagation formulas and simulation results. The model is developed for a linear chain of processes, but extensions for covering also branched and looped product systems are made and described.

Results. The paper proposes a combined use of approximation formulas and Monte Carlo simulation for calculating uncertainty in LCAs, developed primarily for the sequential approach. During the calculation, a parameter observation controls the performance of the approximation formulas. Quantitative threshold values are given in the paper. The combination thus transcends drawbacks of simulation and approximation.

Conclusions and Outlook. The uncertainty question is a true jigsaw puzzle for LCAs and the method presented in this paper may serve as one piece in solving it. It may thus foster a sound use of uncertainty assessment in LCAs. Analysing a proper management of the input uncertainty, taking into account suitable sampling and estimation techniques; using the approach for real case studies, implementing it in LCA software for automatically applying the proposed combined uncertainty model and, on the other hand, investigating about how people do decide, and should decide, when their decision relies on explicitly uncertain LCA outcomes – these all are neighbouring puzzle pieces inviting to further work.

Keywords: Approximation formula; error propagation; life cycle inventory (LCI); life cycle impact assessment (LCIA); models; Monte Carlo Simulation; quantitative thresholds; uncertainties

1 Uncertainties in LCAs

The term 'uncertainties' describes the fact that measured values frequently do not match the true values, but differ from them in a probabilistic manner. In a more narrow sense, uncertainties are probabilistic errors of quantitative values.

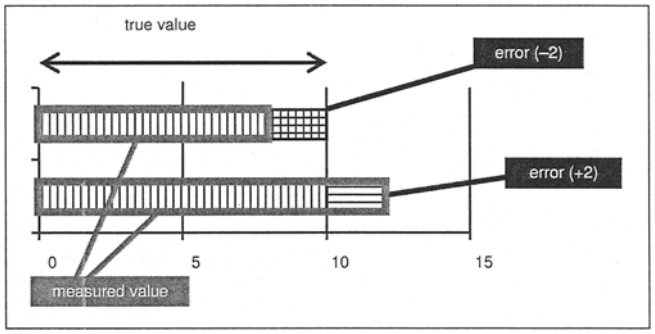

Fig. 1: True value, measured value, and error

An error is defined as the difference between the measured value and the true value (**Fig. 1**).

This definition may also be expressed in a simple formula (**Eq. 1**):

$$\Delta x = x - x_t, \text{ or } x = x_t + \Delta x \quad (1)$$

with

Δx: Error in x
x: Observed / measured value for variable x
x_t: True value for x

Given that the true value is unknown for any measurement, it is not possible to exclude uncertainty in measured values completely or, in other words, to obtain error-free, certain values. This holds true also for quantitative data used in Life Cycle Assessments (LCAs). Therefore, quantitative data used in LCAs (material flows, etc.) is also uncertain.

If a calculation is performed based on several data points which may be uncertain, the uncertainties propagate through the system, show in the calculated result and, provided there are any, also show in intermediate results, for the LCA e.g. in the inventory. This propagation is not easy to foresee, large uncertainties may result from small input uncertainties, while on the other hand, uncertainties from different input values may diminish each other.

For identifying uncertainties, the single random error is not of much interest, since it will differ in every calculation due to its random nature. Interesting is the 'average' or 'mean' error that is likely to appear. For a random error of a variable x, this 'standard mean error' is calculated as the stand-

ard deviation s of the random errors Δx_i, obtained from a series of calculations [19, p. 2], [20, p. 45] (Eq. 2):

$$s = \sqrt{\frac{1}{m-1}\sum_{i=1}^{m}(x_i - x_t)^2} \quad (2)$$

with

x_i: Observed value for variable x
x_t: True value for x
m: Number of calculations performed

For practical calculations, it is common to use the mean $\bar{x}$ as an estimate for the (unknown) true value [20, p. 45]:

$$x_t \cong \bar{x} = \frac{1}{m}\sum_{i=1}^{m} x_i$$

If a calculation is performed with uncertain data, and the uncertainty is disregarded, then a decision based on the calculation result may be made completely in the wrong direction [32, p. 48].

Considerable work has been done in the field of uncertainties and LCAs. Uncertainty is addressed in LCA data formats [3–4,14]; uncertainty is commonly seen as an indicator for data quality [5–6]. Several approaches try to reduce the uncertainty in LCA results by identifying hot spots for uncertainty propagation, both in the calculation and in data used [7–9].

In an abstract way, the 'uncertainty problem' in LCAs can be seen as a simple input/output box (**Fig. 2**), with three sub-problems:

- Problem 1: Assessing errors in input data;
- problem 2: Assessing the propagation of errors in the calculation;
- problem 3: Assessing errors in the calculation's outcome, interpreting outcomes with inherent errors and uncertainties

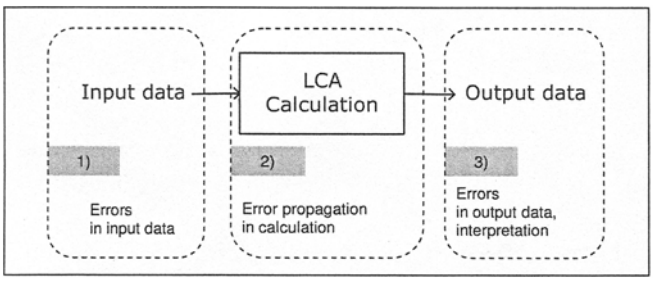

Fig. 2: The calculation of an LCA, seen as an Input/Output box, and three sub-problems related to the assessment of quantitative errors

2 Approaches of Uncertainty Estimation in LCAs: Strengths and Weaknesses

There are a number of proposals on how to describe the error propagation in a calculation. Proposals include interval calculation [10], fuzzy logic approaches (e.g. [11]), Gaussian error propagation formulas [12], and Monte Carlo Simulation [7,9,15].

A Monte Carlo Simulation uses a simple yet ingenious procedure (e.g. [21]): It basically varies input data of the calculation according to a given probability distribution, runs the calculation, and stores the outcome / output data of the calculation. This procedure must be repeated often enough (typically 10,000 times) in order to be sure to obtain input values that adequately represent the selected probability distribution.

For a function f(x) = y, one run of the simulation may be expressed as in **Eq. 3**:

$$f(x) = f(x_t + \Delta x) = y_t + \Delta y = y \quad (3)$$

with (for other symbols, see Eq. 1):

Δy: Error in y
y: Observed / calculated value for variable y
y_t: True value for y

Performing the calculation without introducing errors gives the undisturbed, 'true values' for the model, while performing the calculation with errors in input data yields output data with inherent errors. Subtracting both results gives the true error in the outcome (see Eq. 1). By calculating the standard deviation from the outcome y (see Eq. 2), a reasonable estimate for the 'average' true error, the mean standard error, may be obtained.

Thus, the Monte Carlo Simulation is able to reveal true uncertainty in the calculation's result. In other words, the Monte Carlo Simulation fits well for part 2 of the uncertainty problem (see Fig. 2). However, the simulation cannot correct ill-specified input uncertainties (part 1 of the uncertainty problem), and it does not tell what to do with the uncertainty it calculates (part 3 of the uncertainty problem).

As a drawback, simulating all input parameters can easily exceed available computer resources and time, for a complete and extensive LCA. For this reason, a common approach is to select parameters from all possible parameters, and to simulate uncertainties only for those parameters that have, by expert judgement, an assumed relevance for the uncertainty in the outcome. This approach is implemented in LCA software [34, pp 96].

On the other side, all other methods mentioned above have the problem that, especially under unfavourable conditions (non-linearity in the calculation, relatively high random errors), the uncertainty estimate given by approximation formulas can deviate largely from the 'true uncertainty'. In addition, a control of their estimate is difficult, since 'true errors' and 'true values', and thus also the true value for the uncertainty in the outcome, are not known.

In summary, both simulation and approximation suffer from a problem in uncertainty calculation: The simulation is able to give a reasonable estimate for the uncertainty, but in the common application only for previously selected parameters that are assumed to be most relevant; the approximation can only approximate the uncertainty, and it frequently cannot give a reasonable estimate.

3 An Uncertainty Estimation Model for LCAs

3.1 Aim of the model, and principal approach for building it

Aim of the model. The Model has the aim (1) to investigate error propagation / uncertainties in LCAs, and (2) to find an approach for calculating the uncertainties in an LCA. In addition, this approach should be efficient – as short in computing time as possible, as moderate in computing resource demands as possible – it should provide estimates for the uncertainty as correct as possible, and it should be applicable in practice.

The model has to recognise the specific drawbacks of approximation and simulation (see section 2), that apply also for LCAs. Specifically, the approach should be able to deal with uncertainty in all parameters of a study (not only in selected parameters), and it should provide 'parameters' that indicate how well different formulas perform in calculating the uncertainty. These parameters should be measurable in a real case study. Ciroth [16] developed the basic model in a doctoral thesis.

Principle approach: A virtual LCA case study. A straightforward approach would be to analyse an existing case study for uncertainties and error propagation, to implement different algorithms for uncertainty calculation for this case study, and to compare and analyse the results. However, in an LCA case study, as in any quantitative data, the true value for random errors, and thus the uncertainty, is an inherent property of data. It is not directly 'visible' or measurable (see also section 1). For this reason, true values for uncertainties cannot be measured from a case study, they can only be estimated. However, these true values are essential to decide how well different algorithms are able to calculate the uncertainty (aim 2, above), and they are important to investigate the uncertainty propagation in LCAs (aim 1). In addition, one single case study does not allow a variation of all different parameters of an LCA (some parameters are fixed for the study) – which is important for the analysis of uncertainty propagation, and for finding suitable parameters.

For both of these reasons, a different concept has been used instead:

- We implemented a model that calculates an LCA, including approximation formulas and simulation.
- Within the model, a virtual case study was created with the aim of coming as close to a realistic case study as possible. Data in the case study was generated according to probability distributions for true values and random errors.
- This case study was analysed by means of explorative data analysis in order to find relevant parameters.
- A subsequent step investigated relevant parameters in order to condense the number of parameters further, and to identify threshold values for the remaining parameters. In this step, parameters for single calculation steps of an LCA were varied systematically, overwriting case study data.

Fig. 3 shows the sequence of interlinked calculations implemented in the model. The model covers inventory calculation, impact assessment calculation, and some possible elements of valuation (normalisation, and a calculation of the difference between two scenarios).

Uncertainty calculation methods in the model. For each of the calculation steps, the model contains calculation and approximation formulas derived in Ciroth's thesis [16]. From several approximation formulas investigated, the following two were most promising:

- The 'error propagation formula', originally by Gauss, in which the random error (or uncertainty) in a calculated figure y, $y = f(x_1, x_2, \ldots, x_M)$ is given by the sum of the partial 1st order derivatives of this function, multiplied by their uncertainty estimates. Both derivative and input uncertainties are squared, and a square root is extracted from the whole sum (Eq. 4):

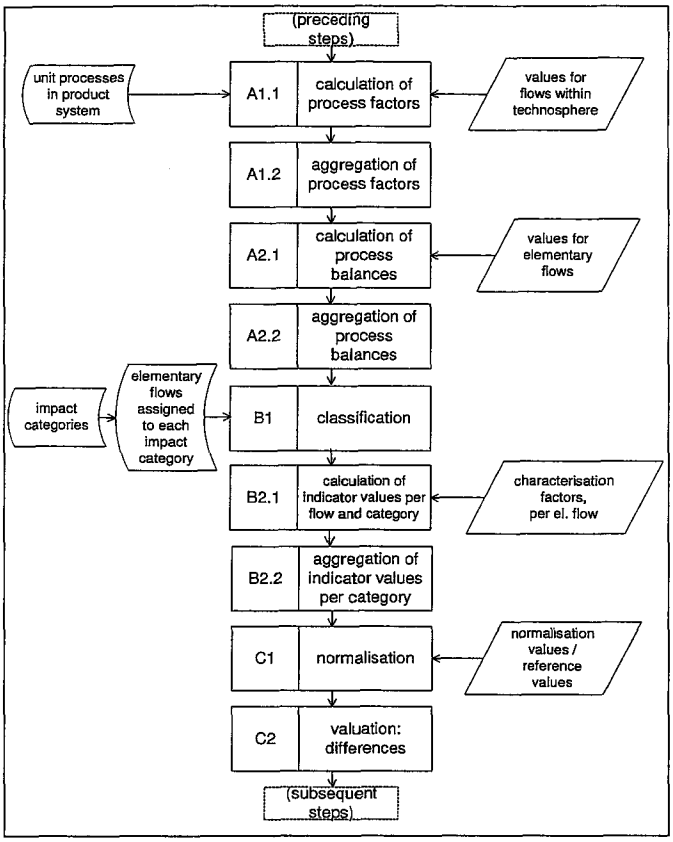

Fig. 3: The calculation of an LCA as a sequence of calculation steps, as implemented in the model, with qualitative and quantitative data input. Capital / number pairs (A1.1) refer to elements of the model

$$s_y = \sqrt{\sum_{k=1}^{M} \left(\frac{\partial f}{\partial x_k}\right)^2 (s_{x_k})^2} \qquad (4)$$

with

s_y: Random error estimate in y (calculation result);
s_{x_k}: Random error estimate in x_k (input value);

$\frac{\partial f}{\partial x_k}$: Partial derivative of the calculation function f.

- A higher order extension of the Gaussian error propagation formula, given by Bader and Baccini [18, pp 108] (Eq. 5):

$$s_y = \sqrt{(s^{(1)}{}_y)^2 + (s^{(2)}{}_y)^2} \qquad (5)$$

with

$(s_y^{(1)})^2 = \sum_{k=1}^{M} \left(\frac{\partial f}{\partial x_k}\right)^2 (s_{x_k})^2$ → the Gaussian formula for error propagation, see Eq. 4

and $(s_y^{(2)})^2 = \sum_{k,l=1}^{M} \left[\frac{1}{4}\frac{\partial^2 f}{(\partial x_k)^2}\frac{\partial^2 f}{(\partial x_l)^2} + \frac{1}{2}\left(\frac{\partial^2 f}{(\partial x_k \partial x_l)}\right)^2 + \frac{\partial f}{\partial x_k}\frac{\partial^3 f}{\partial x_k (\partial x_l)^2}\right](s_{x_k})^2 (s_{x_l})^2$ → the extension of the Gaussian formula.

Note that the Bader / Baccini extension spans matrices of the input values' derivatives and uncertainty estimates, with indices k and l for columns and lines, respectively, and M as the number of input values.

These results from approximation formulas are compared to results obtained from a Monte Carlo Simulation, which is able to provide estimates for the true uncertainties (see also section 2), also within the LCA model, and may hence serve as a reference.

Algorithm used in the model for calculating the inventory, impact assessment, and valuation. Algorithms for the calculation of LCAs are not generally accepted, and frequently overlooked in LCA papers [27, p. 3]. There are at least two different ways for calculating an LCA: A 'network approach', also called 'sequential method' [27, p. 100], which can be extended to a Petri net calculation [23–26], on the one hand, and a matrix approach [12,27], on the other.

The matrix approach solves a linear equation system simultaneously, and, according to Heijungs and Suh, requires the performance of a matrix inversion for solving the linear equation system As = f [27, pp 20].

A number of different algorithms exist for inverting a matrix. For practical applications, Heijungs and Suh recommend an implementation of different inversion algorithms for LCAs to be able to choose from and compare to each other [27, p. 198].

However, performing a matrix division is sufficient for solving the linear equation system, and numerically preferable[1] [13,29–30]. For performing a matrix division, generally the Gauss elimination algorithm, either in its pure form or in optimised modifications like the LU decomposition, is preferred [30]. Both matrix division and matrix inversion require a square matrix; in cases where it is not possible to have a square matrix A, a solution may be obtained by minimising the norm of the difference vector $||As-f||_2$, following the method of Gaussian squares [40].

The network approach, on the other hand, typically starts at one 'central' process; from this process, flows exchanged with other processes are calculated (amount of flows for products and pre-products); the connected processes need then be scaled with a linear factor, the scaling factor, according to their contribution to the product system; process balances (typically taken from a database) are multiplied with the scaling factor for each process; the aggregation of all process balances in the product system is the inventory. Flows that are exchanged between processes vanish in the inventory, which ideally contains only elementary flows. These elementary flows are classified into impact categories, multiplied by characterisation factors specific for each category, and the products are finally aggregated within each impact category. The result is an indicator value per impact category [22, pp 593], [16, pp 17].

The approximation formulas used in this paper refer to the sequential approach. Formulas used in the model are given in the **Appendix**.

3.2 Virtual case study, EDA, and systematic analysis

The **virtual case study** was built in the following manner: Its product system consisted of two linear chains of 52 processes each (**Fig. 4**, n = 52), one chain for upstream and one for downstream processes. Some processes appear multiple times in the product system, which corresponds to processes like transport via truck, in LCA studies. In summary, there are 60 unique processes in the system. Processes in the process chain were grouped in these unique processes by assigning uniformly distributed numbers between 1 and 60 to them. As a result, every process appears between one and five times in the process chain. Every process has a true value for one input flow in its balance, linking to one other process, and for one output flow, linking to one other process. These values were obtained following a normal distribution. With the exception of the utmost processes in the chain, each process is linked to two other processes via product flows, and each of the process balances contains between 3 and 20 elementary flows. The number of elementary flows per unique process follows a uniform distribution. Ten different impact categories were calculated. Via random number generation, each category was assigned between 3 and 30 elementary flows of the overall 495 elementary flows, following a uniform distribution. The resulting structure of the model (processes with flows and types of elementary flows, elementary flows assigned to impact categories) was not changed in the simulations.

In addition, each product flow – flow between processes – and elementary flow was assigned a normally distributed true value, and a normally distributed error. True values for

[1] A citation from the Matlab Function reference: "In practice, it is seldom necessary to form the explicit inverse of a matrix [by inv]. A frequent misuse of inv arises when solving the system of linear equations. One way to solve this is with x = inv(A)*b. A better way, from both an execution time and numerical accuracy standpoint, is to use the matrix division operator x = A\b. This produces the solution using Gaussian elimination, without forming the inverse" [29].

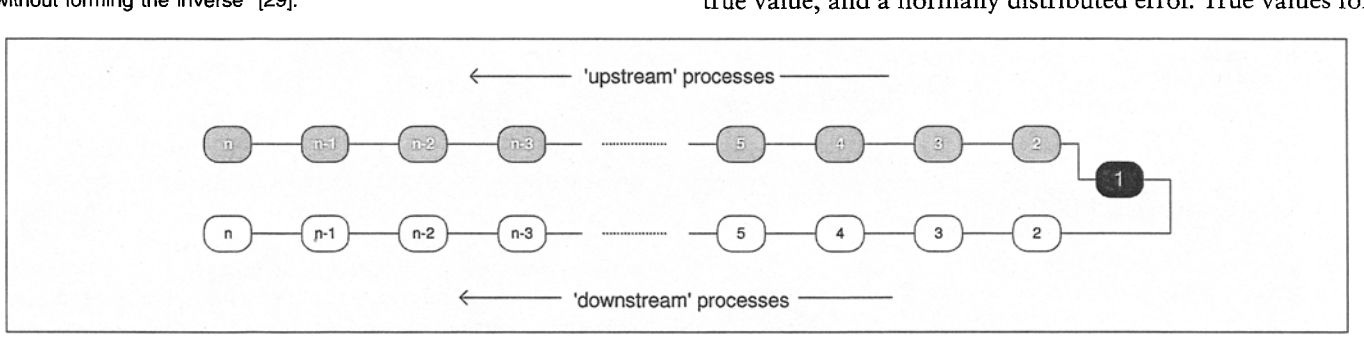

Fig. 4: The structure of the process system in the model – a linear chain of processes

characterisation factors and for reference values used in the normalisation (step C1) were generated following a normal distribution. In the simulation, we varied the error to get variations in the true value or, in terms of Eq. 3, we varied Δx to obtain variations in y, and thus, via the standard deviation in y, the Monte Carlo Simulation's estimate for the uncertainty in y. In every case, the probability distribution of the error in the simulation remains the same as in the initial generation of the error. Each Monte Carlo Simulation took 10,000 runs. For example, impact category '7' was assigned elementary flows 49, 77, 123, and 147, and elementary flow 147 has a true value of 3.76 and an error of about 0.45. The latter, the error, was varied in the simulations according to the probability distribution described above. For further details of the virtual case study, please refer to [16].

As a result, the case study model consists of a linear chain of processes; it calculates an LCA from inventory to impact assessment to normalisation and scenario comparison. Errors in quantitative data which it uses are kept apart from true values, and both errors and true values are observable in every calculation step.

Explorative Data Analysis EDA. Aim of the analysis is to explore the virtual case study for relevant patterns and parameters, which indicate whether uncertainty estimates given by approximation formulas are close to those of a Monte Carlo Simulation. The analysis thus follows the idea of exploratory data analysis (EDA)[2]. We plotted, sorted and clustered the data calculated for each step separately, and for the whole LCA calculation procedure as well. It helped that the virtual case study provided a broad range of parameter constellations, including unplanned ones, due to the random generation of input data. Investigated parameters comprise the amount of flows exchanged between processes (flows in technosphere), the number and amount of all elementary flows and of all characterisation factors, the amount of the reference values used in normalisation; in fact, the number of times each process appears in the process system, for both true values and errors, and the position of each process in the chain, i.e. if the calculated uncertainty is based on input data alone (process 1 in Fig. 4) or comes with a high degree of estimated uncertainty based on previous processes.

Main findings of the EDA were:

- The absolute amount of the true value, and of the true error, has a high influence on the performance of the approximation formula.
- The Gauss formula is always more vulnerable to high uncertainties than the Bader Baccini formula.
- A problem in the application of approximation formulas is that they need to build upon the uncertainty they calculate in preceding calculation steps. Since they do not manage to provide a correct estimate, the following estimate gets worse. Thus, there is a 'meta error propagation' (error propagation in calculating the random error) for approximation formulas, and the error in the error calculation is an important parameter for the performance of the formula in a next calculation step.
- Some of the multitude of parameters appeared to have more relevance for uncertainty propagation and calculation. For step A1.1, for instance, the input uncertainty, the absolute amount of flows, the relative uncertainty in these flows, and the length of the process chain were more relevant by far than other parameters.

Systematic analysis of selected parameters. In a second step, we tested each of those parameters more thoroughly that appeared to be relevant in the previous analysis. To find thresholds for each relevant parameter where the approximated uncertainty starts to deviate from the Monte Carlo uncertainty, the value of each parameter was varied continuously within intervals. As in the virtual case study, values from the approximation formulas were compared to simulation results. The analysis was no longer based on each single LCA calculation step, but on types of calculation steps, grouping those with identical formulas and similar parameters, such as step A2.1 and B2.1 for example.

The analysis served to condense the number of relevant parameters. Some of the parameters investigated proved not to be relevant themselves, but mere indicators of other parameters or effects. For example, the number of processes preceding a process seemed, in the virtual case study, a highly important parameter for step A1.1. A result of the systematic analysis is, in fact, that the error in the calculated input uncertainty is the only relevant parameter, see the following section for a specific example.

3.3 Results of the model analysis

The two-step procedure provided results for every calculation step shown in Fig. 3. As an example, **Fig. 5** shows results for A1.1, the calculation of scaling factors, from the systematic analysis, with input uncertainty and the length of the process chain as varied parameters. For small input uncertainties (relative uncertainty 0.01, Fig. 5a), nearly no difference between the compared methods is visible. Raising the input uncertainty to 0.1 changes the true uncertainty from a digressive shape to a linear function (see also the fitted functions in the boxes in Fig. 5). The Bader / Baccini formula now also produces a linear function, but with a lower slope, while Gauss still has a digressive function (see Fig. 5b). Consequently, calculated uncertainty estimates differ considerably with a longer process chain, and the difference for the Gauss formula is higher. With a relative input uncertainty of 0.2, the picture changes dramatically; the Monte Carlo Simulation uncertainty now has a shape that is best met by a third degree polynomial, with extremely high values for relative uncertainty of up to 70, while both approximation formulas do not change the shape of their plots (see Fig. 5c).

For interpreting these figures, the following thoughts are useful: For each process k, the input uncertainty can be separated (i) into the uncertainty in parameters of process k, namely the uncertainty in the flows to other processes $mOutput_k$ and $mInput_k$, and (ii) into the uncertainty in the scaling factor p of the preceding process k-1. Within the parameter values analysed for the results, the propagating uncertainty of the scaling factors is far more important for the uncertainty of the scaling factor of process k than the

[2] Explained by Tukey as "exploratory data analysis is detective work. [...] restricting one's self to the planned analysis – failing to accompany it with exploration – loses sight of the most interesting results too frequently to be comfortable" [36, p. 3].

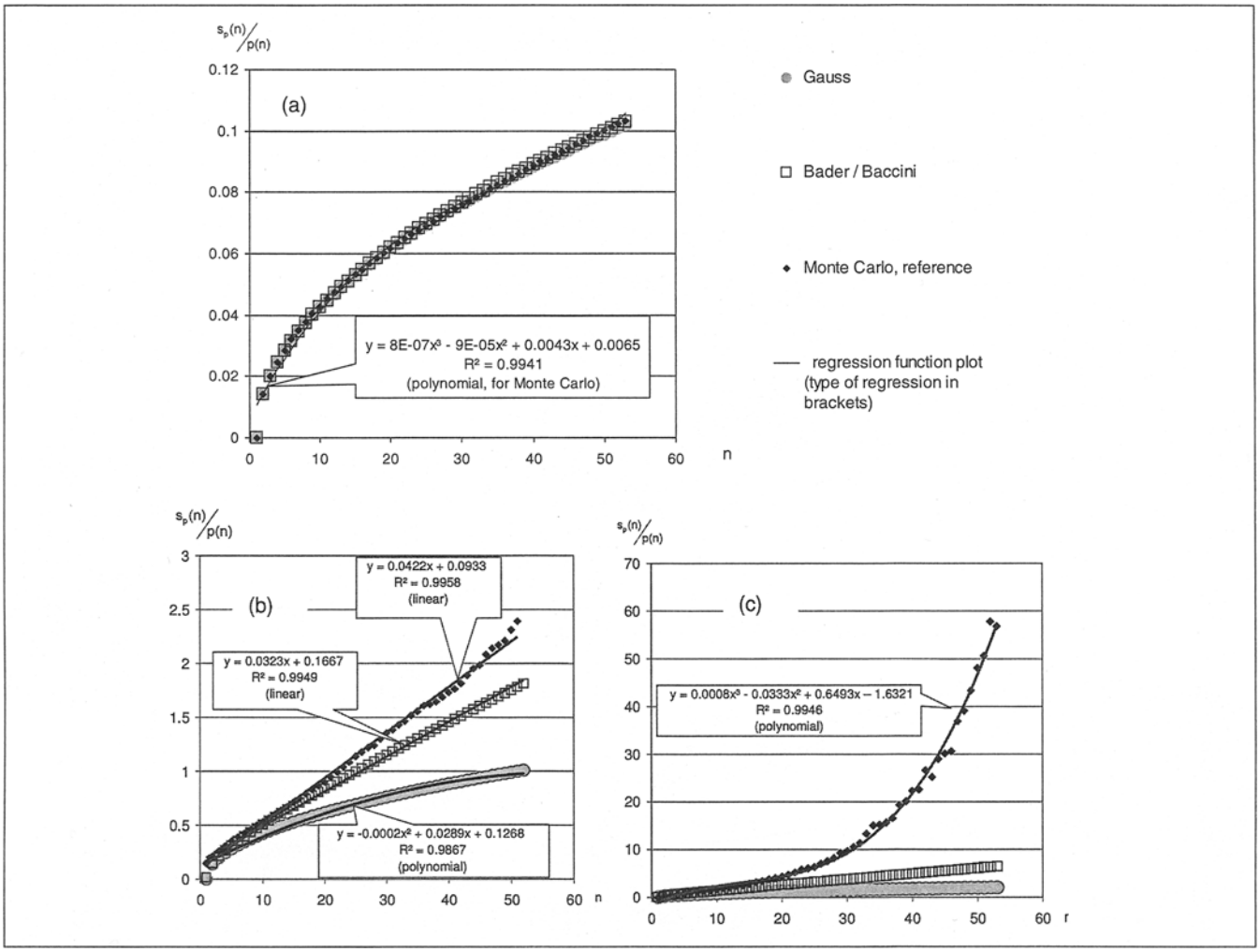

Fig. 5: Results for the calculation of the process scaling factors, with relative uncertainty in input parameters of 0.01 (Fig. 5a), 0.1 (Fig. 5b), and 0.2 (Fig. 5c) with: n: number of each process in the process chain; $s_p(n)/p(n)$: relative uncertainty in scaling factor for process n

parameter uncertainty. For process n = 50 in Fig. 5b, for example, the propagating relative uncertainty (uncertainty from process 49) is 2.27, and the parameter uncertainty (from $mOutput_{50}$) is 0.1. In Fig. 5a, the propagating uncertainty for process 50 is 0.10, the input uncertainty is 0.01, and the Gauss formula gives a reasonable uncertainty estimate. Obviously, the number of processes in the chain is not directly important for the performance of the approximation formula, since it is identical for both cases in process 50. Important is the relative input uncertainty, expressed as the relative uncertainty of the scaling factor of the preceding process. If the relative input uncertainty here is below 0.2, the Gaussian approximation formula may be used. If it is below 0.8, the Bader / Baccini formula gives a reasonable estimate for the uncertainty, and above, the Monte Carlo simulation should be used. The thresholds can be read from the plots of the results. In addition, we consulted the calculated data, and additional plots of the relative difference of approximation formula and determined the relative uncertainty where the approximation formula result equals the simulation result, with a tolerance of 10%, which we rounded down to be on the safe side. **Fig. 6a** and **6b** correspond to the example shown in Fig. 5b. It plots the relative difference for Gauss (Fig. 6a) and for Bader Baccini (Fig. 6b) over the relative uncertainty, with a relative input uncertainty of 0.1 (the same as in Fig. 5b). It also shows the 10% interval and, in the thesis, it motivated the 20% threshold for Gauss (the interval is exceeded at a relative uncertainty if 0.26, which is rounded down to 0.2), and the 80% threshold for Bader (here the interval is left at 0.8 which is then also the threshold). Note that the simulation does not have a function as a result. The regression curves are only estimates with no more information than plots; in Fig. 5b, for example, regression curves for Gauss and Monte Carlo simulation do not meet at all (this holds also if the real Gauss approximation formula is taken, which is not shown in the plot). Hence, the way to interpret plots instead of analysing functions is the most direct one; it allows one to deal with the raw, non-modified data.

For every calculation step in the model, the relative uncertainty (in statistical expression: the variation coefficient) turns out to be the determining parameter for the goodness of fit of the uncertainty as calculated by the approximation formulas.

A second major result is that both approximation formulas manage, within the model, to calculate the uncertainty cor-

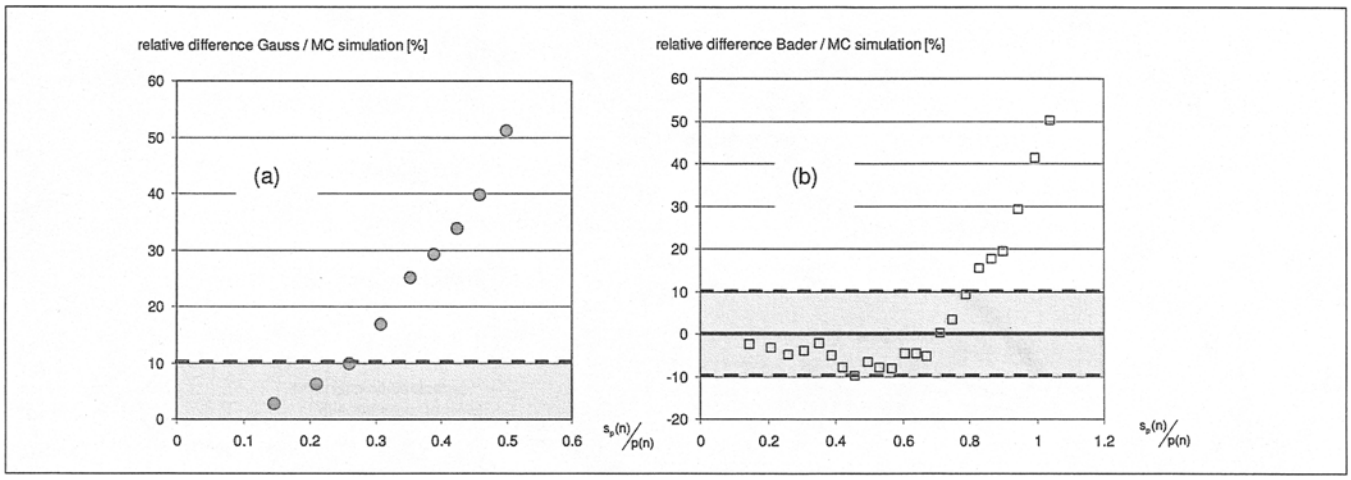

Fig. 6: Relative difference for Gauss (6a) and for Bader Baccini (6b) over relative uncertainty, for A1.1, with a relative input uncertainty of 0.1 (the same as in Fig. 5b). $s_p(n)/p(n)$: relative uncertainty in scaling factor for process n; the relative difference is calculated; for Gauss as (s p(n)Monte Carlo – sp(n)Gauss) / sp(n)Monte Carlo *100%, and for Bader as (s p(n)Monte Carlo – sp(n)Bader) / sp(n)Monte Carlo *100%

rectly for every calculation step, unless the relative error that they calculate does not exceed certain thresholds. The Bader / Baccini formula generally has higher thresholds. Those calculation steps that use only addition and substraction (A1.2; A2.2; B2.2; C2), always give a reasonable estimate for uncertainty, hence there are no thresholds. For the other calculation steps, a threshold was determined similar to the example given above. **Table 1** summarises the thresholds for each calculation step. Note that thresholds need to be applied on the calculation result, see **Fig. 7**, which generalises the thoughts above on the calculation of process scaling factors.

More sensitive calculation steps generally have lower thresholds. The results summarised in Table 1 allow recommending an **approach for estimating uncertainties in LCA results:**

Start the calculation with approximation formulas, Gaussian or, preferable if implemented, the Bader / Baccini formula,

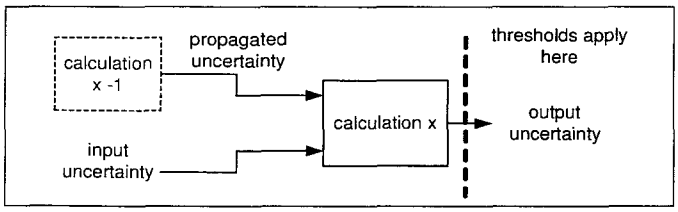

Fig. 7: Input uncertainty, propagated uncertainty, output uncertainty, and the application of the thresholds, per calculation. 'Calculation' means: Calculation step (see Fig. 3), or also sub-steps within a, sequentially calculated, calculation step, like the process scaling factors in step A1.1

and check during the calculation for thresholds according to Table 1. If these thresholds are exceeded, either change from Gauss to Bader / Baccini or from Bader / Baccini to a Monte Carlo Simulation, which may also be used 'piecewise', for single calculation steps, or even for sections within one calculation step, e.g. for a part of the process chain.

3.4 Consequences of a branched and looped product system

The model as described in the previous chapters is based on a linear chain of processes as the simplest possible form of a product system. Practical examples for product systems are usually more complex, they include branches of process chains and loops of processes. How does a more complicated structure affect the interpretations gained from the model?

Fig. 8 shows a branched chain of processes (left) and a loop within a process chain (right). The boxes represent processes; the sequential calculation starts from process 1 and continues to higher numbers.

As indicated in the figure, a branched chain can be split into two separate linear chains, which are identical before the branching process (in the picture process number 3). Calculation and results for each of the branches that are now directly connected to the 'stem' are similar to the linear chain of processes represented in the model. The stem (processes 1 to 3 in Fig. 8, on the left) needs to be calculated only once since it is identical in both cases. As a conclusion, it is not

Table 1: Thresholds for the use of approximation formulas for estimating the uncertainty in the calculation steps of an LCA

Calculation step	Uncertainty calculation method: recommended use until		
	Gauss	Bader / Baccini	Monte Carlo
A 1.1	Variation coefficient in input scaling factor < 20%	Variation coefficient in input scaling factor < 80%	No threshold
A1.2	No threshold	No threshold	No threshold
A2.1	Variation coefficient in scaling factors and in elementary flows < 40%	Variation coefficient in scaling factors and in elementary flows < 100%	No threshold
A2.2	No threshold	No threshold	No threshold
B2.1	Variation coefficient in characterisation factors and aggregated elementary flows in inventory < 40%	Variation coefficient in characterisation factors and aggregated elementary flows in inventory < 100%	No threshold
B2.2	No threshold	No threshold	No threshold
C1	Variation coefficient in nominator < 400% Variation coefficient in denominator < 60%	Variation coefficient in nominator < 400% Variation coefficient in denominator < 1000%	No threshold
C2	No threshold	No threshold	No threshold

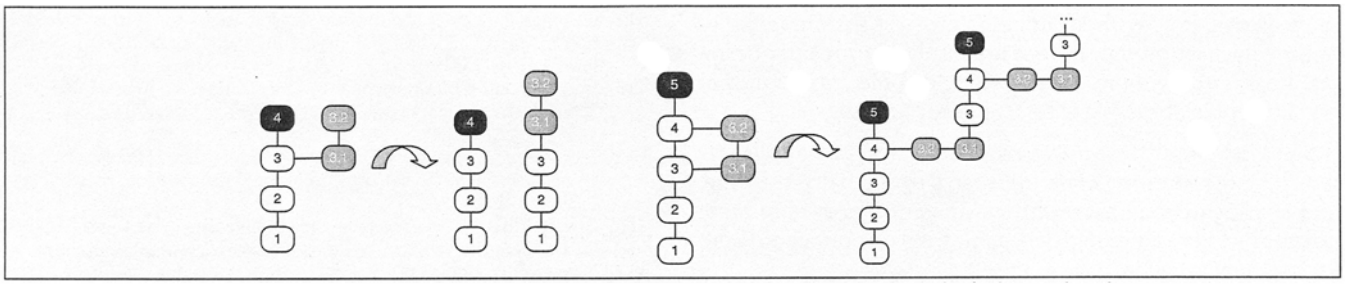

Fig. 8: Illustration for the linearization of branches (left) and loops (right) in the process chain, further explanation see text

necessary to split branches into separate linear chains for the calculation, since they can be calculated in the same way as a single linear chain or stem. Therefore, results for the linear chain are also valid for branched process chains.

The case of loops in the process chain is more complicated. The figure shows that a loop can be written as a sequence of branches. Unlike a normal branch, these branches contain many identical processes. Another point to consider is the convergence of the loop, which is essential for being able to calculate any reasonable uncertainty figure. The calculation is hence not completely trivial, and would need more space for an explanation. Basically, the system may be calculated as a branched system, and results for processes that appear multiple times (e.g. process 4 in the linearised system in Fig. 8) are obtained by adding the results for each of the appearances of the process. Recent tests show that the thresholds provided in Table 1 are valid also in this case [2].

3.5 Applicability and limitations of model and results

Results of the paper are the threshold values, the recommended approach for uncertainty estimation, and the approach of a virtual uncertainty case study.

Threshold values are developed for a linear chain of processes, but recent findings show that they also apply for more complex product system structures, see section 3.4. They depend to some degree on the calculation algorithm employed in the model. Both sequential and matrix algorithms may be closer than it seems at first glance; the Gaussian Elimination algorithm, for example, requires a repetitive calculation of quotients of elements of matrix A, which is similar to the quotients needed for process scaling factor calculation in the sequential approach, and the term 'matrix inversion' is sometimes, by Heijungs and Suh as well, used as a synonym for solving the equation system (without meaning to calculate the inverse of the matrix) [13]. At present, the threshold values are limited to the sequential approach. Changing the probability distribution from normal to log normal did not change threshold values in Ciroth's thesis [16]. The model does not explicitly cover correlation among data. Data correlations may be relevant if they concern variables used simultaneously in the calculation. For covering uncertainty, a correlative term needs to be added to the approximation formulas, and the correlation needs to be considered in the random number generation in the simulation. If strong correlation between data exists, an influence on the thresholds seems probable.

The recommended approach for uncertainty estimation does not depend on specific threshold values, but it is most interesting for high thresholds, i.e. in cases where the approximation formulas are able to provide sound estimates.

The approach of investigating uncertainties by the use of a virtual case study, and by the use of explorative data analysis and a subsequent systematic analysis serves to condense the large number of parameters, to study the behaviour of calculation algorithms under uncertainty, and allows one to set true values, and true (random) errors. It is applicable for other types of systems, algorithms (matrix algorithms, or damage functions in impact assessment), and data distributions. The use of virtual data is recognised and well accepted in other scientific fields like biological modelling (e.g. [39]), while it has not yet been used often for LCAs. An example is [38].

4 Conclusions

Integrating approximation formulas and a Monte Carlo Simulation approach in a model that calculates an LCA based on virtual data enables to separate clearly true values and errors, to analyse properly the uncertainty propagation in an LCA calculation, and to investigate how well, and under different conditions, the approximation formulas cope with estimating the propagated uncertainty.

The calculated relative uncertainty, the variation coefficient, turns out to be the parameter that dominantly determines the performance of the approximation formulas (Gauss, Bader and Baccini) in the model. The paper presents quantitative thresholds for the variation coefficient, for each calculation step of an LCA, and for the Gauss and the Bader and Baccini formula. If an approximation formula calculates a variation coefficient below these thresholds, the model shows that the estimate of uncertainty given by the formula is good, and equal to a simulation result. Consequently, approximation formulas can use the variation coefficient to self-check their estimate of uncertainty. It is possible to control the goodness of the estimate during the calculation, which was one of the aims of the model, see section 3.

It is then straightforward to combine approximation formulas and the Monte Carlo approach in the uncertainty calculation, combining their benefits without the downsides (which are a questionable correctness for the approximation formulas, and, sometimes, an excessive demand for time and computer resources for the simulation). This combination is proposed in further detail in the paper.

The model and approach limits itself to the 'technical side' of uncertainty, namely the calculation, disregarding where the uncertainty stems from. It calculates a linear chain of processes, which can be broadened to branched and even

looped structures without impeding the results obtained, as outlined in section 3.4. It does not explicitly consider correlated data; applying its results, thus, has the condition that no such correlation exists to a relevant amount.

If the latter condition holds, results seem broadly applicable for LCA calculations. The approach to virtually calculate and investigate the uncertainty propagation proved to be of utmost benefit and convenience, and we recommend its use for other, similar problems.

Several possible paths for further work come in sight, including an analysis on how to manage the input uncertainty, including suitable sampling and estimation techniques, using the approach for real case studies, and implementing it in LCA software for automatically applying the proposed, combined uncertainty model. For broadening the applicability, investigating correlations in data and their impact on the results has promise, and further exploring the shortly mentioned approaches for relevance estimation in Life Cycles will give light to an often-asked practical question. Investigating for an optimal calculation code for LCAs, also regarding the different sensitivity to uncertainty, the differing condition number of the algorithms, would bring more light in the proper design and selection of a calculation algorithm. Finally, on the output side of the uncertainty problem (see Fig. 2), an analysis of best ways to manage uncertainty in the outcome of an LCA is necessary, for fostering a sound decision support, and for possibly lowering psychological barriers of stakeholders, decision makers, LCA experts and practitioners in dealing with uncertainty in LCAs. And last but not least, we should further explore possible reasons for the occurrence of uncertainty in LCAs.

Acknowledgement. The authors thank PD Dr. Peter Benner from the department of numerical mathematics, Technical University Berlin, for recent advice on numerical issues dealt with in this paper. The paper benefited gratefully from constructive comments of two reviewers.

References

[1] Kahnemann D, Slovic P, Tversky A (Eds) (1982): Judgement under Uncertainty, Heuristics and Biases. Cambridge University Press
[2] Ciroth A (2003): Uncertainty calculation for LCI data: Reasons for, against, and an efficient and flexible approach for doing it. Presentation, International Workshop on Quality of LCI Data, 20–21 October 2003, Forschungszentrum Karlsruhe <http://www.lci-network.de/cms/content/cache/offonce/pid/468>
[3] <http://www.spold.org/>
[4] <http://www.globalspine.com/LCA/>
[5] Weidema B, Wesnaes MS (1996): Data quality management for life cycle inventories – An example for using data quality indicators. J Cleaner Prod 4 (3–4) 167–174
[6] Huijbregts MAJ, Norris G, Bretz R, Ciroth A, Maurice B, von Bahr B, Weidema B, de Beaufort ASH (2001): Framework for Modelling Data Uncertainty in Life Cycle Inventories. Int J LCA 6 (3) 127–131
[7] Huijbregts MAJ (2001): Uncertainty and variability in environmental life-cycle assessment. Academisch Proefschrift, Univ. van Amsterdam, Amsterdam
[8] Ciroth A (2000): A new model for the propagation of errors in LCA, shown in a case study. Platform Presentation, 8th SETAC case study symposium, Brussels
[9] Canter KG, Kennedy DJ, Montgomery DC, Keats JB, Carlyle WM (2002): Screening Stochastic Life Cycle Assessment Inventory Models. Int J LCA 7 (1) 18–26
[10] Le Téno JF (1999): Visual Data Analysis and Decision Support Models for Non-Deterministic LCA. Int J LCA 4 (1) 41–47
[11] Pohl Chr, Ros M (1996): Sind Ökobilanzen zu präzise? In: Intelligente Methoden zur Verarbeitung von Umweltinformation. 2. Bremer KI-Pfingstworkshop (Tagungsband), metropolis, Marburg, pp 121–136
[12] Heijungs R (1996): Identification of key issues for further investigation in improving the reliability of life-cycle assessments. J Cleaner Prod 4 (3–4) 159-166
[13] Sangwon Suh (2003): personal communication, 11 June 03
[14] Ecospold data format <http://www.ecoinvent.org/en/pdf/DocumentationEcoSpold1.2.pdf>
[15] McCleese D, LaPuma P (2002): Using Monte Carlo Simulation in Life Cycle Assessment for Electric and Internal Combustion Vehicles. Int J LCA 7 (4) 230–236
[16] Ciroth A (2001): Fehlerrechnung in Ökobilanzen. Dissertation TU Berlin 2001 <http://edocs.tu-berlin.de/diss/2001/ciroth_andreas.htm>
[17] Bevington R, Robinson DK (1992): Data Reduction and Error Analysis for the Physical Sciences. WCB/McGrawHill, Boston
[18] Bacchini P, Bader HP (1996): Regionaler Stoffhaushalt. Spektrum Akademischer Verlag, Heidelberg Berlin Oxford
[19] Wolf H (1979): Ausgleichungsrechnung II. Ferd. Dümmlers Verlag, Bonn
[20] Höpcke W (1980): Fehlerlehre und Ausgleichsrechnung, Walter de Gruyter, Berlin New York
[21] Vose D (1996): Quantitative Risk Analysis: A Guide to Monte Carlo Simulation Modelling. John Wiley & Sons, Chichester, New York, Brisbane, Toronto, Singapore
[22] Fleischer G, Schmidt WP (1995): Life cycle assessment. In: Ullmann's Encyclopedia of Industrial Chemistry, 5th Ed, Vol B8, Weinheim: VCH Verlagsgesellschaft, pp 585–600
[23] Schmidt M (1996): Die Modellierung von Stoffrekursionen in Ökobilanzen. In: [24, pp 97–117]
[24] Schmidt M, Schorb A (1996): Stoffstromanalysen in Ökobilanzen und Öko-Audits. Springer Verlag, Berlin Heidelberg
[25] Möller A, Rolf A, Page B, Wohlgemuth V (2001): Foundations and Applications of Computer Based Material Flow Networks for Environmental Management. In: Rautenstrauch C Patig S (Eds) Environmental Information Systems in Industry und Public Administration. Hershey, PA USA, pp 379–396
[26] Ciroth A (1998): Beispielhafte Anwendung der Iterativen Screening-Ökobilanz. Master thesis, Fachgebiet Abfallvermeidung der TU Berlin, Berlin
[27] Heijungs R, Suh S (2002): The computational Structure of Life Cycle Assessment. Kluwer Academic Publishers, Dordrecht
[28] Hertwich EG, Hammitt JK (2001): A Decision-Analytic Framework for Impact Assessment, Part 2: Midpoints, Endpoints, and Criteria for Method Development. Int J LCA 6 (5) 265–272
[29] Matlab Function Reference; Function 'inv', Version 6.5, Release 13, 2002
[30] Golub G, Ortega JM (1993): Scientific Computing. Academic Press
[31] Ciroth A (2000): Einfluß der Reihenfolge von Bezugsszenariorechnung, Normierung und Charakterisierung auf das Ergebnis der LCA. In: Umweltschutz im neuen Jahrhundert, Festschrift anlässlich des 60. Geburtstages von Univ.-Prof. Dr.-Ing. Günter Fleischer, TK Verlag, Neuruppin, pp 95–104
[32] Ross St, Evans D, Webber M (2002): How LCA Studies Deal with Uncertainty. Int J LCA 7 (1) 47–52
[33] Plinke E, Schonert M, Meckel H, Detzel A, Giegrich J, Fehrenbach H, Ostermayer A, Schorb A, Heinisch J, Luxenhofer K, Schmitz St, Umweltbundesamt (Hrsg) (2000): Ökobilanz für Getränkeverpackungen II, Hauptteil. UBA Texte 37/00, Berlin 2000, p 133: 'Eine spezielle Methode zur Fehlerabschätzung in Ökobilanzen wurde bislang nicht entwickelt' (a specific method for estimating uncertainties in LCAs is not developed yet)
[34] GaBi 4 Reference manual, 2003-01-29
[35] Björklund AE (2002): Survey of Approaches to Improve Reliability in LCA. Int J LCA 7 (2) 62 – 72
[36] Tukey JW (1977): Exploratory Data Analysis, Addison Wesley, Reading
[38] Ackermann R (1997): Ableitung von allgemein nutzbaren Sachbilanzmodulen – Untersuchungen am Beispiel der Aufbereitung von Altkunststoffen aus dem Dualen System. Dissertation TU Berlin
[39] Omlin M, Reichert P (1999): A comparison of techniques for the model prediction uncertainty. Ecological Modelling 115, 45–59
[40] Björk A (1996) Numerical Methods for Least Squares Problems. SIAM, Society of Industrial and Applied Mathematics, Philadelphia
[41] Hartung J, Elpelt B (1989): Multivariate Statistik. Oldenbourg, München
[42] Laux H (1998): Entscheidungstheorie. Springer Verlag, Berlin
[43] Kammen DM, Hassenzahl DM (1999): Should We Risk It? Exploring Environmental, Health and Technology Problem Solving, Princeton University Press, Princeton NJ
[44] Morgan MG, Henrion M (1990): Uncertainty – A guide to dealing with uncertainty in quantitative risk and policy analysis. Cambridge University Press, NY

Received; July 25th, 2003
Accepted: June 12th, 2004
OnlineFirst: June 14th, 2004

Appendix: Calculation formulas and approximation formulas used

Note: The Bader / Baccini extension uses the general term

$$s_y = \sqrt{\left(s^{(1)}{}_y\right)^2 + \left(s^{(2)}{}_y\right)^2},$$

the first term under the root being the Gauss approximation: $s^{(1)}y = sy^{Gauss}$. In the following, only the second term under the square root, the extension $s^{(2)}y$, is given to avoid repetitions, if not mentioned otherwise.

Step	
A1.1	**Explanation and reference:** For a chain of processes, the calculation starts at a 'central' process, and scales the linear factors of the other processes in the chain according to that process, processes before the central process ('pre') and processes after the central process ('post') [22, pp. 405] [16].
	Calculation formula $p_{n,d} = \prod_{i=1}^{n} \frac{mOutput_{i-1}}{mInput_i}$ $\qquad$ $p_{n,u} = \prod_{i=1}^{n} \frac{mInput_{i-1}}{mOutput_i}$ with – p_n: Scaling factor of process n; mOutput$_i$: Output exchange of process I; mInput$_i$: Input exchange of process i
	Gauss $s^{Gauss}_{p_{n,d}} = \sqrt{\left[\prod_{i=1}^{n} \frac{mOutput_{i-1}}{mInput_i}\right]^2 \cdot \sum_{i=1}^{n}\left[\left(\frac{s_{mOutput_{i-1}}}{mOutput_{i-1}}\right)^2 + \left(\frac{s_{mInput_i}}{mInput_i}\right)^2\right]}$ $s^{Gauss}_{p_{n,u}} = \sqrt{\left[\prod_{i=1}^{n} \frac{mInput_{i-1}}{mOutput_i}\right]^2 \cdot \sum_{i=1}^{n}\left[\left(\frac{s_{mInput_{i-1}}}{mInput_{i-1}}\right)^2 + \left(\frac{s_{mOutput_i}}{mOutput_i}\right)^2\right]}$
	Bader / Baccini $s^{(1)}{}_y = s^{Gauss}_{p_n}$, for both types of processes, 'pre' and 'post'. $s^{(2)}{}_y = \sqrt{\sum_{k=1}^{M}\sum_{l=1}^{M}(a_{kl}{}^2 \cdot s^2_{mOutput_k} \cdot s^2_{mOutput_l} + b_{kl}{}^2 \cdot s^2_{mOutput_k} \cdot s^2_{mInput_l} + c_{kl}{}^2 \cdot s^2_{mInput_k} \cdot s^2_{mOutput_l} + d_{kl}{}^2 \cdot s^2_{mOutput_k} \cdot s^2_{mOutput_l})}$ with a_{kl}, b_{kl}, c_{kl}, d_{kl} being elements of square matrices A, B, C, D, with indices k for columns and l for lines, respectively. with – n: the number of the process in the process chain; M: the total number of parameters, for this process The matrices themselves are defined as follows: – for 'downstream' processes: $a_{kl} = \begin{cases} \frac{\sqrt{2}}{2}(p_{dn}) \cdot \frac{1}{mOutput_k \cdot mOutput_l}, k \neq l \\ 0, k = l \end{cases}$ $\qquad$ $b_{kl} = \frac{\sqrt{2}}{2}(p_{dn}) \cdot \frac{\sqrt{5}}{mOutput_k \cdot mInput_l}$ $c_{kl} = \frac{\sqrt{2}}{2}(p_{dn}) \cdot \frac{1}{mInput_k \cdot mOutput_l}$ $\qquad$ $d_{kl} = \begin{cases} \frac{\sqrt{2}}{2}(p_{dn}) \cdot \frac{\sqrt{10}}{mInput_k \cdot mInput_l}, k \neq l \\ \frac{\sqrt{2}}{2}(p_{dn}) \cdot \frac{3\sqrt{2}}{mInput_k \cdot mInput_l}, k = l \end{cases}$ – for 'upstream' processes: $a_{kl} = \begin{cases} \frac{\sqrt{2}}{2}(p_{un}) \cdot \frac{\sqrt{10}}{mOutput_k \cdot mOutput_l}, k \neq l \\ \frac{\sqrt{2}}{2}(p_{un}) \cdot \frac{3\sqrt{2}}{mOutput_k \cdot mOutput_l}, k = l \end{cases}$ $\qquad$ $b_{kl} = \frac{\sqrt{2}}{2}(p_{un}) \cdot \frac{1}{mOutput_k \cdot mInput_l}$ $c_{kl} = \frac{\sqrt{2}}{2}(p_{un}) \cdot \frac{\sqrt{5}}{mOutput_l \cdot mInput_k}$ $\qquad$ $d_{kl} = \begin{cases} \frac{\sqrt{2}}{2}(p_{un}) \cdot \frac{1}{mInput_k \cdot mInput_l}, k \neq l \\ 0, k = l \end{cases}$
A1.2	**Explanation and reference:** A unit process may appear several times in the product system (take e.g. the unit process for electricity generation). It is more efficient to aggregate the scaling factors for these processes before multiplying them with the process balances, which is done in the subsequent step A2.1 [16]. The aggregation needs not care for the location of the process in the process system, a distinction between 'pre' and 'post' processes is no longer necessary.
	Calculation formula $p_n = \sum_{m=1}^{k} p_{nm}$ with – p_n: Scaling factor of process n; p_{nm}: Scaling factor of the m-th occurrence of process n

Step	
	Gauss $s_{p_n}^{Gauss} = \sqrt{\sum_{m=1}^{k} \left(s_{p\,nm}\right)^2}$
	Bader / Baccini In this step, the Bader Baccini extension $s^{(2)}y$ is zero since higher derivates of the calculation formula are zero.
A2.1, A2.2	**Explanation and reference:** The linear, aggregated scaling factors for each process are multiplied with the process balances, which contain elementary input and output flows (step A2.1). The product is then aggregated per elementary flow over all processes. Result is the inventory balance of the product system, with entries for each elementary flow (step A2.2) [16].
	Calculation formula $m_i = \sum_{n=1}^{k} p_n m_{ni}$ with – p_n: Scaling factor of process n; m_{ni}: Elementary flow i in process n; m_i: Elementary flow i
	Gauss $s_{m_i} = \sqrt{\sum_{n=1}^{k} \left[\left(m_{ni} \cdot s_{p_n}\right)^2 + \left(p_n \cdot s_{m_{ni}}\right)^2\right]}$
	Bader / Baccini $s^{(2)}{}_{m_i} = \sqrt{\sum_{k=1}^{M}\sum_{l=1}^{M} \frac{1}{2}\left(\left(s_{p_l}\right)^2\left(s_{p_k}\right)^2 + 2\left(s_{p_l}\right)^2\left(s_{m_{ki}}\right)^2 + \left(s_{m_{li}}\right)^2\left(s_{m_{ki}}\right)^2\right)}$
B1	**Classification is a logical assignment without calculation formulas.**
B2.1; B2.2	**Explanation and reference:** Elementary flows as result from the inventory calculation are multiplied with applicable characterisation factors for obtaining indicator values in impact categories (step B2.1), in step B2.2, these indicator values per category are aggregated. [16]
	Calculation formula $k_j = \sum_i m_i c_{ij}$ with – k_j: Indicator value in impact category j; c_{ij}: Indicator value in category j for elementary flow m_i
	Gauss $s_{k_j}^{Gauss} = \sqrt{\sum_i \left(c_{ij} \cdot s_{m_i}\right)^2 + \left(m_i \cdot s_{c_{ij}}\right)^2}$
	Bader / Baccini $s^{(2)}{}_{k_j} = \sqrt{\sum_{k=1}^{M}\sum_{l=1}^{M} \frac{1}{2}\left(\left(s_{c_{lj}}\right)^2\left(s_{c_{kj}}\right)^2 + 2\left(s_{c_{lj}}\right)^2\left(s_{m_k}\right)^2 + \left(s_{m_l}\right)^2\left(s_{m_k}\right)^2\right)}$
C1	**Explanation and reference:** The normalisation step sets the aggregated indicator values per impact category in relation to reference values [16], [31]
	Calculation formula $z_j = \frac{k_j}{k_{j\,ref}}$ with – z_j: Normalised indicator value in category j; k_j: Indicator value in category j; $k_{j\,ref}$: Reference indicator value for category j
	Gauss $s_{z_j}^{Gauss} = \sqrt{\left(\frac{s_{k_j}}{k_{j\,ref}}\right)^2 + \left(\frac{k_j \cdot s_{k_{j\,ref}}}{\left(k_{j\,ref}\right)^2}\right)^2}$
	Bader / Baccini $s^{(2)}{}_{z_j} = \sqrt{a^2 \cdot s_{k_j}^4 + 2 \cdot b^2 \cdot s_{k_j}^2 \cdot s_{k_{j\,ref}}^2 + c^2 \cdot s_{k_{j\,ref}}^4}$ with $a = \frac{\sqrt{2}}{\left(k_{j\,ref}\right)^2} \qquad b = \frac{\sqrt{2}}{2 \cdot \left(k_{j\,ref}\right)^2} \qquad c = \frac{3k_j}{k_{j\,ref}{}^3}$
C2	**Explanation and reference:** This step simply subtracts the normalised results of one scenario from those of another scenario, per impact category [16], [31]
	Calculation formula $z_{Delta\,j\,a} = z_{j\,a} - z_{j\,ref}$ with – $z_{Delta\,j\,a}$: Difference of scenario a and reference scenario, for category j; z_{ja}: Normalised indicator result for category j, scenario a; $z_{j\,ref}$: Normalised indicator result for category j, reference scenario
	Gauss $s_{z_{Delta\,i\,a}} = \sqrt{\left(s_{z_{ia}}\right)^2 + \left(s_{z_{i\,ref}}\right)^2}$
	Bader / Baccini In this step, the Bader Baccini extension $s^{(2)}y$ is zero since higher derivates of the calculation formula are zero.